

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

SPRÁVA A MONITORING ZAŘÍZENÍ ROUTEROS V MALÝCH SÍTÍCH

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ONDŘEJ LEŠTINSKÝ

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

SPRÁVA A MONITORING ZAŘÍZENÍ ROUTEROS V MALÝCH SÍTÍCH

MANAGEMENT AND MONITORING OF ROUTEROS DEVICES IN SMALL NETWORKS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ONDŘEJ LEŠTINSKÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JOSEF HÁJEK

BRNO 2011

Abstrakt

Tato bakalářská práce popisuje návrh a implementaci informačního systému, který dokáže vzdáleně konfigurovat zařízení se systémem RouterOS. Hlavní funkcí systému je správa klientů, kteří se mohou připojit do počítačové sítě prostřednictvím v systému evidovaných zařízení. U jednotlivých klientů je možné omezit rychlost připojení a zobrazit jejich aktuální stav. Všechna evidovaná zařízení je možné zobrazit na mapě. Správu systému mohou vykonávat jen autorizovaní správci.

Abstract

This bachelor's thesis describes the design and implementation of an information system that can remotely configure the device running RouterOS. The main features of system is the administration of clients who can connect to a network through a system of registered devices. All clients connection speed can be limited and viewed their current status. All registered devices can be displayed on the map. The management system can perform only authorized administrators.

Klíčová slova

Mikrotik, RouterOS, RouterBoard, Správa klientů, Informační systém, Google Maps

Keywords

Mikrotik, RouterOS, RouterBoard, Client management, Information System, Google Maps

Citace

Ondřej Leštinský: Správa a monitoring zařízení RouterOS v malých sítích, bakalářská práce, Brno, FIT VUT v Brně, 2011

Správa a monitoring zařízení RouterOS v malých sítích

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Josefa Hájka.

.....
Ondřej Leštinský
12. května 2011

Poděkování

Tímto bych velice rád poděkoval vedoucímu této bakalářské práce panu Ing. Josefu Hájkovi za velice vstřícný přístup a odborné rady, které mi při tvorbě práce pomohly.

© Ondřej Leštinský, 2011.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	3
2 MikroTik RouterOS	4
2.1 MikroTik	4
2.2 RouterOS	4
2.3 RouterBoard	4
2.4 Konfigurační nástroje	6
2.4.1 Aplikace Winbox	6
2.4.2 SSH/telnet konzolové připojení	6
2.4.3 Webové rozhraní Webbox	7
2.4.4 API (application programming interface)	7
2.5 Základní konfigurace	7
2.5.1 IP adresy	8
2.5.2 Uživatelské účty	8
2.5.3 MAC autentizace	8
2.5.4 Čas a datum	8
2.5.5 Povolení API	8
2.6 Generování statistik	8
3 MikroTik API	9
3.1 Protokol	9
3.1.1 Příkazy	9
3.1.2 Argumenty	9
3.1.3 Dotazy	10
3.1.4 Odpovědi	10
3.2 Dostupné API	10
3.3 API PHP třída	10
4 Návrh informačního systému	11
4.1 Informační systém	11
4.2 Webová aplikace	11
4.3 Požadované funkce	12
4.3.1 Správa uživatelů	12
4.3.2 Správa ROS zařízení	13
4.3.3 Správa připojených klientů	13
4.3.4 Zobrazení statistiky	14
4.3.5 Propojení s mapovým podkladem	14
4.4 E-R diagram	15

5	Implementace navrženého systému	16
5.1	Zvolený programovací jazyk	16
5.2	Adresářové schéma	16
5.3	Generování obsahu HTML stránek	17
5.4	Databáze	18
5.4.1	Schéma relační databáze	18
5.4.2	Přístup do databáze	18
5.5	Konfigurace ROS	19
5.6	Zobrazení na mapě	20
5.6.1	Statické mapy	20
5.6.2	Použití v systému	21
5.7	Implementace funkcí systému	21
5.7.1	Uživatelská obsluha	21
5.7.2	Interakce s RB	24
5.8	Metriky kódu	29
6	Závěr	30
6.1	Demonstrace navrženého systému	30
6.2	Zhodnocení systému	31
6.3	Vylepšení do budoucna	31
A	Obsah CD	35

Kapitola 1

Úvod

Téměř v každé počítačové síti je třeba rozlišit různé segmenty sítě, ať už z bezpečnostního hlediska nebo kvůli přehlednosti. Z tohoto důvodu je pak síť pomocí směrovačů rozdělena na podsítě. Takových podsítí může být v síti velký počet a tedy i značný počet směrovačů. V těchto případech je pak velice komplikované kontrolovat přístup oprávněných uživatelů do sítě. Jejich správa a monitorování může být ve velkém počtu poněkud náročná.

Cílem této práce je vytvořit informační systém, díky kterému by správci sítě mohli pohodlně z jednoho centrálního místa monitorovat a částečně konfigurovat všechna zařízení zajišťující z převážné části bezdrátový přístup uživatelů do sítě. V této práci se bude jednat o zařízení firmy MikroTik používající RouterOS.

Celkem obsahuje tato práce šest kapitol. Ve druhé kapitole bych rád představil firmu MikroTik, objasnil základní pojmy používané v průběhu práce a nastínil možnosti pro konfiguraci RouterOS. V následující části popíši konfiguraci pomocí API (rozhraní pro programování aplikací). Další kapitoly zaměřím na návrh informačního systému a jeho implementaci. Součástí těchto kapitol bude seznam funkcí systému včetně popisu vzdálené správy a propojení s mapovým podkladem.

V poslední kapitole uvedu jednoduchý demostrační příklad a následně zhodnotím implementovaný systém. V samotném závěru ještě zmíním funkce, o které by popisovaný systém mohl být v budoucnu rozšířen.

Kapitola 2

MikroTik RouterOS

Tato kapitola slouží k základnímu seznámení se zařízeními MikroTik. Jsou vysvětleny pojmy používané ve zbytku práce. Uveden je také základní výčet funkcí v závislosti na druhu licence. Na závěr této kapitoly se podíváme na základní způsoby konfigurace a uvedu několik příkladů jednoduché konfigurace. Informace obsažené v této kapitole jsou čerpány ze stránek společnosti Mikrotik [8].

2.1 MikroTik

Jedná se o název firmy, která se zaměřuje na bezdrátové systémy pro poskytovatele internetového připojení (ISP). Firma byla založena roku 1995 v Lotyšsku. Od té doby vyrábí a dodává na světový trh své produkty. Mezi největší odběratele patří firmy jako Siemens, Hewlett-Packard či NASA.

2.2 RouterOS

RouterOS (dále jen ROS) je operační systém založený na Linuxu používaný v zařízeních firmy MikroTik (nejčastěji RouterBoardy). Systém však může být nainstalován i na klasickém PC. První verze tohoto routerového operačního systému byla vyvinuta v roce 1997. V době psaní této práce je aktuální verze 5.

Použití ROS je rozděleno na licenční stupně. Celkem existuje šest stupňů licencí, liší se počtem použitelných funkcí a samozřejmě jejich cenou. V tabulce 2.1 je uveden základních výčet funkcí a jejich návaznost na druh licence.

2.3 RouterBoard

RouterBoard (dále jen RB) je deska plošných spojů osazená procesorem, pamětmi a dalšími součástkami. Jedná se tedy o malou základní desku. Vybavená je navíc sériovým RS232 rozhraním a alespoň jedním ethernetovým rozhraním, respektive mini pci slotem. Do mini pci slotu lze připojit bezdrátový adaptér s anténou, RB pak může fungovat jako wifi stanice či přístupový bod. RB může být použit jako router, switch, firewall nejčastěji ale jako výše zmíněný přístupový bod. Nabízena je také možnost pro FTP přístup.

Pro představu čtenáře je uvedena tabulka 2.2 s hardwarovými vybavením vybraných typů.

Level	0	1	3	4	5	6
Označení	FREE	DEMO	WISP CPE	WISP	WISP	Controller
Wireless AP	24h limit	–	–	ano	ano	ano
Wireless Client and Bridge	24h limit	–	ano	ano	ano	ano
RIP, OSPF, BGP protocols	24h limit	–	ano (jen RB)	ano	ano	ano
EoIP tunnels	24h limit	1	neomezeně	neomezeně	neomezeně	neomezeně
PPPoE tunnels	24h limit	1	200	200	500	neomezeně
PPTP tunnels	24h limit	1	200	200	500	neomezeně
L2TP tunnels	24h limit	1	200	200	500	neomezeně
OVPN tunnels	24h limit	1	200	200	neomezeně	neomezeně
VLAN interfaces	24h limit	1	neomezeně	neomezeně	neomezeně	neomezeně
HotSpot active users	24h limit	1	1	200	500	neomezeně
RADIUS client	24h limit	–	ano	ano	ano	ano
Queues	24h limit	1	neomezeně	neomezeně	neomezeně	neomezeně
Web proxy	24h limit	–	ano	ano	ano	ano
Synchronous interfaces	24h limit	–	–	ano	ano	ano
User manager active sessions	24h limit	1	10	20	50	neomezeně
Možný upgrade	–	–	ROS v6.x	ROS v6.x	ROS v7.x	ROS v7.x
Podpora	–	–	–	15 dní	30 dní	30 dní

Tabulka 2.1: Licence RouterOS [12]

RB	Rychlost CPU [Mhz]	Velikost RAM [MB]	Architektura	Počet LAN	Počet miniPCI	Počet USB
RB411	300	32	MIPS-BE	1	1	0
RB433	300	64	MIPS-BE	3	3	0
RB493G	680	256	MIPS-BE	9	3	1
RB800	800	256	PPC	3	4	0

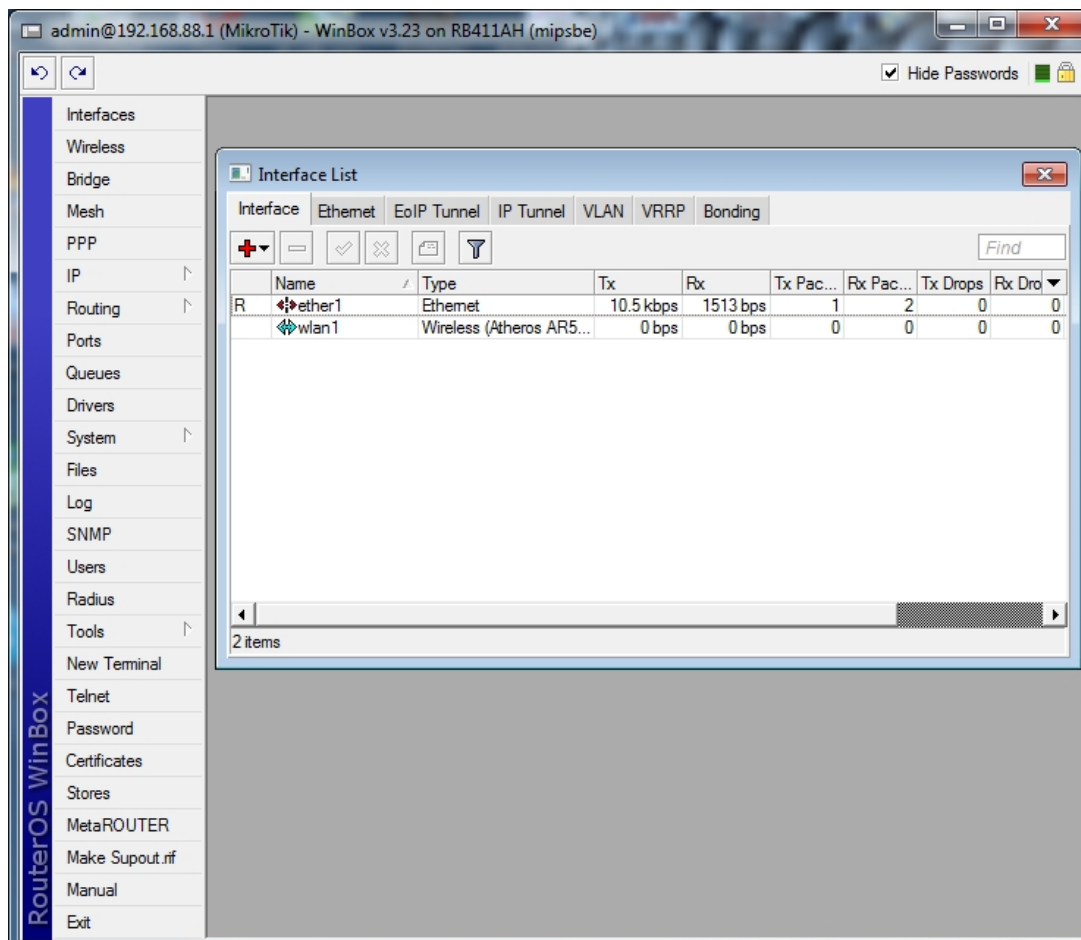
Tabulka 2.2: Vybavení RouterBoardů [1]

2.4 Konfigurační nástroje

Existují čtyři základní způsoby jak ROS nakonfigurovat.

2.4.1 Aplikace Winbox

Aplikace umožňuje pohodlně přes grafické uživatelské rozhraní konfigurovat ROS. Po spuštění Winboxu aplikace automaticky vyhledá pomocí MNDP (MikroTik Neighbor Discovery Protocol) a CDP (Cisco Discovery Protocol) připojitelná zařízení. Po zadání uživatelského jména a hesla se umožní správa ROS (Obr. 2.1).



Obrázek 2.1: Aplikace Winbox.

2.4.2 SSH/telnet konzolové připojení

Konfigurace přes SSH a telnet je stejná. Jediný rozdíl, je v tom, že pomocí SSH se používá šifrovaný přenos dat. Po připojení se dostaneme do intuitivního menu, ve kterém se můžeme pohybovat pomocí příkazů. Příkazy snadno zjistíme vyvoláním nápovědy, stisknutím „?“. Pro usnadnění práce je možno použít klávesu „tab“, která dokončuje rozepsané příkazy.

2.4.3 Webové rozhraní Webbox

Rozhraní se vyvolá po zadání IP adresy zařízení do adresové řádky internetového prohlížeče (Obr. 2.2). Po přihlášení jsou dostupné všechny základní funkce pro konfiguraci, které nabízí Winbox.



Obrázek 2.2: Webové rozhraní Webbox.

2.4.4 API (application programming interface)

Díky API lze přidat do našeho systému jen ty funkce, které opravdu využijeme. Můžeme si tedy vytvořit vlastní aplikaci na způsob Winboxu. Tento způsob konfigurace bude využit při tvorbě finální aplikace. API je věnovaná celá následující kapitola.

2.5 Základní konfigurace

Základní konfigurace je jednoduchá a zařízení lze nastavit během pár minut. Jako příklady jsou uvedeny posloupnosti příkazů pro konfiguraci pomocí SSH připojení. Příklady zde uvedené budou jen v lehké pozměněné podobě využity i ve finální aplikaci.

2.5.1 IP adresy

Nastavení IP adres se provádí v menu `ip/address`. Příklad nastavení adresy `10.10.10.1` s maskou `255.255.255.0` na rozhraní `wlan1`:

```
/ip address add address=10.10.10.1 netmask=255.255.255.0 interface=wlan1
```

2.5.2 Uživatelské účty

Aby zařízení mohlo konfigurovat více uživatelů, je nutné vytvořit i více uživatelských účtů v menu `user`. Vytvoření uživatelského účtu `uzivatel` s heslem `heslo` a plnými oprávněními pro správu zařízení:

```
/user add name=uzivatel password=heslo group=full
```

2.5.3 MAC autentizace

V menu `interface/wireless/access-list` je možné povolit přístup pro stanice, které mohou se zařízením komunikovat. Je však nutné nastavit autentizaci v bezpečnostním profilu. Povolení přístupu pro MAC adresu `AA:BB:CC:DD:EE:FF` k bezdrátovému rozhraní `wlan1`:

```
/interface wireless access-list add interface=wlan1 mac-address=AA:BB:CC:DD:EE:FF
```

2.5.4 Čas a datum

Na zařízení je také možné v menu `system/clock` nastavit čas a datum:

```
/system clock set time=hh:mm:ss date=mesic/den/rok
```

2.5.5 Povolení API

API používá port `8728`, který implicitně není aktivován. Povolení API:

```
/ip service enable api
```

2.6 Generování statistik

Podle [9] ROS nabízí jediný vestavěný nástroj, který umožňuje ve zvolené časové periodě zaznamenávat různé statistiky a zobrazit je formou grafu. Jedná se o nástroj umístěný v menu `tool/graphing`. Díky němu lze sledovat a zobrazit následující grafy:

- stav RB
- využití zdrojů (CPU, RAM)
- provoz procházející přes rozhraní
- provoz procházející přes fronty

Shromážděná data z routeru se do grafu promítnou vždy po 5 minutách, ukládaná na disk jsou však vždy v nastavené periodě. Přístup ke grafům lze omezit jen na uživatele vybraných podsítí. Statistiku jsou dostupné přes webové rozhraní v sekci `Graphs`. Pro detailnější informace o zařízení lze použít nástroj **Cacti**. Pro komplexnější sledování sítě nabízí MikroTik aplikaci **The Dude**. Oba tyto nástroje využívají pro sledování protokol SNMP.

Kapitola 3

MikroTik API

V první části této krátké kapitoly se blíže seznámíme s protokolem MikroTik API a ukážeme si jeho základní principy sloužící ke komunikaci. Ve druhé části budou zmíněny možnosti využití API v programovacích jazycích. My si blíže představíme PHP třídu využitou při implementaci výsledného systému. Všechny uvedené informace v této kapitole jsou převzaty ze stránky věnované MikroTik API [\[11\]](#).

3.1 Protokol

Protokol je formátován jako proud přenášených slov, každé slovo je zakódováno podle své délky. Z jednotlivých slov jsou tvořeny tzv. věty. Každá věta je ukončena slovem nulové délky. Komunikace je založená na principu dotaz – odpověď. Podle začátku každé věty lze rozlišit, zda se jedná o dotaz či odpověď. Celkem existují následující druhy vět:

3.1.1 Příkazy

Věta začíná „/“, za tímto znakem následuje příkaz. Příkazy jsou stejné, jako při zadávání do konzole, rozdíl však spočívá v tom, že mezery mezi jednotlivými příkazy jsou nahrazeny dalšími „/“. Pro API jsou navíc přidány následující příkazy:

- cancel
- listen
- getall
- login

Příklad získání informací o všech ethernetových rozhraních:

```
/interface/ethernet/getall
```

3.1.2 Argumenty

K tomu, aby bylo možné příkazy blíže specifikovat, je nutné použít argumenty. Každá věta s argumentem začíná „=“. Poté následuje jméno argumentu, další znak „=“ a požadovaná hodnota. Na pořadí při zadání argumentů nezáleží. Příklad nastavení jména rozhraní:

```
=name=wlan1
```

3.1.3 Dotazy

Pokud je třeba vyhledat nebo vyfiltrovat některé údaje, je nutné větu začít „?“. Pak už věta pokračuje stejně jako u zadání argumentů, tedy jménem argumentu, rovnítkem a vyhledávanou hodnotou. Dotaz je vyhodnocován pro každou položku zvlášť. Pokud položka nevyhovuje, je ignorována, v opačném případě je předána dále ke zpracování. Příklad vypsaní všech klientů aktuálně připojených k rozhraní wlan1:

```
/interface/wireless/access-list/print  
?interface=wlan1
```

3.1.4 Odpovědi

Aby komunikace fungovala, je nutné na výše uvedené věty odpovědět. K odpovědi slouží věty začínající „!“ . Následuje typ odpovědi a parametry. Rozlišujeme tyto odpovědi:

- **!done** — zpráva o dokončení provedeného příkazu
- **!trap** — indikuje chyby a vyjíměčné situace
- **!re** — požadavek o další data

3.2 Dostupné API

MikroTik na své webové stránce věnované API zveřejňuje volně k dispozici zdrojové texty pro práci s ROS. Jde o velký počet programovacích jazyků, jmenovitě například Delphi, Java, C, PHP a další. Je nutné ale dodat, že tyto zdrojové texty nepocházejí přímo od MikroTiku, byly vytvořeny nadšenci pro ROS. Každá z těchto knihoven obsahuje prostředky pro navázání spojení, odeslání příkazu a čtení odpovědi.

3.3 API PHP třída

Tato PHP třída je pro nás v případě této práce jistě nejzajímavější, zajišťuje komunikaci pomocí jazyka PHP. Jejím autorem je Denis Basta, volně dostupná na [10]. Po vytvoření instance této třídy máme k dispozici několik metod, nejdůležitější jsou zde následně uvedené.

První v pořadí je metoda **connect**, k zavolání této metody je nutné parametry předat IP adresu zařízení, přístupové uživatelské jméno a heslo. Díky této metodě se naváže a uchová spojení se zařízením. Komunikace probíhá na transportní síťové vrstvě (sokety). V případě neúspěchu se metoda opakovaně snaží o navázání spojení a až po několika pokusech je spojení prohlášeno za neúspěšné, mezi jednotlivými pokusy je definován časový interval, proto se může zdát připojování poněkud zdoluhavé. Pokud je spojení navázáno, je nastaven atribut **connected**. Tento atribut je pak vhodné před prvním použitím dalších metod vždy zkontrolovat.

Když je spojení navázané, můžeme metodou **write** odesílat příkazy do zařízení. Pokud potřebujeme odeslat několik příkazů za sebou, je nutné o tom informovat nastavením druhého parametru na hodnotu **false**.

Po odeslání dat obvykle následuje volání metody **read**. Metoda přijme a zpracuje odpověď od zařízení, kterou vrací jako dvourozměrné asociativní pole.

Poslední zde představená je metoda **comm**. Tato metoda je velice užitečná, neboť spojuje předchozí dvě metody. Do prvního parametru se vloží příkaz, který se má vykonat a do druhého se vloží pole atributů. Metoda opět vrací dvourozměrné asociativní pole.

Kapitola 4

Návrh informačního systému

Cílem kapitoly je definovat informační systém, popsat webovou aplikaci a její umístění v rámci počítačové sítě. Hlavní pozornost ale věnujeme výčtu funkcí, od kterého se odvíjí návrh struktury informačního systému. Na základě popsanych funkcí je vytvořen E-R diagram.

4.1 Informační systém

Přesná definice pojmu Informační systém neexistuje a ani ji nelze jednoduše vytvořit, neboť každý uživatel či tvůrce Informačního systému používá různé terminologie a zdůrazňuje jiné aspekty. Můžeme však říci, že Informační systém (IS) lze chápat jako systém vzájemně propojených informací a procesů, které s těmito informacemi pracují. Přičemž pod pojmem procesy rozumíme funkce, které zpracovávají informace do systému vstupující a transformují je na informace ze systému vystupující. Zjednodušeně můžeme říci, že procesy jsou funkce zabezpečující sběr, přenos, uložení, zpracování a distribuci informací. Pod pojmem informace pak rozumíme data, která slouží zejména pro rozhodování a řízení v rozsáhlejších systémech.

Do celkové funkce IS se také promítá nezanedbatelná položka okolí. Okolí informačního systému tvoří veškeré objekty, které změnou svých vlastností ovlivňují samotný systém, a také objekty, které naopak mění své vlastnosti v závislosti na systému.

Celkově tedy můžeme říci, že IS je softwarové vybavení firmy, které je schopné na základě zpracovávaných informací řídit procesy podniku nebo poskytovat tyto informace řídicím pracovníkům tak, aby byli schopni vykonávat řídicí funkce, mezi které patří zejména plánování, koordinace a kontrola veškerých procesů firmy.[17]

4.2 Webová aplikace

Aplikace je podle požadavků navržena a implementována jako webová, a proto je pro běh aplikace nutné v počítačové síti vyčlenit počítač, který plní funkci webového serveru. Tento počítač je v síti umístěn tak, aby z něj byl přístup do všech částí sítě, ve kterých chceme vzdáleně konfigurovat zařízení.

Pro uživatele z toho plyne jednoznačná výhoda, není už třeba instalovat žádný další software pro přístup do systému. Všechny úkony lze provádět pomocí internetového prohlížeče, který je součástí běžné výbavy každého počítače.

4.3 Požadované funkce

Navrhovaný IS musí umožnit správu uživatelských účtů, základní konfiguraci všech dostupných RB a zvládat správu klientů a jejich adres. Užitečné je také zobrazení jednoduchých statistik a vhodné propojení s mapovým podkladem. Navrhovaný systém v sobě musí mít zakomponovanou obsluhu pro všechny funkce uvedené v následujících podkapitolách.

4.3.1 Správa uživatelů

Celý systém je obsluhovaný přihlášenými uživateli. Tito uživatelé jsou ve své podstatě správci sítě a záleží jen na hlavním administrátorovi, jaká oprávnění uživatelům přidělí.

Vstup do systému

Každý uživatel se musí do systému autentizovat pomocí přihlašovacího jména a hesla. Autentizace vždy probíhá na úvodní webové stránce. Informace o uživateli jsou uchovány a použity pro ověření autorizace k dalším činnostem v systému. Po dokončení práce se uživatel ze systému odhlásí.

Vytvoření nového uživatele

Uživatel, který má řádné oprávnění, může vytvořit nového uživatele. Uchováváno je uživatelské jméno, heslo, jméno, příjmení a kontaktní emailová adresa. Po vytvoření je novému uživateli umožněno jen přihlášení do systému.

Kontaktování dalších uživatelů

Každý uživatel má přístup alespoň k emailové adrese ostatních uživatelů.

Editace základních údajů

Všichni přihlášení mohou změnit své vlastní kontaktní údaje a přístupové heslo. Editaci údajů může provádět i oprávněný uživatel.

Zajištění autorizace

Uživatel smí vstoupit jen do té části systému, do které má oprávnění (i přihlášení do systému). Přístupová oprávnění je možné průběžně modifikovat. Stejný princip je použit i pro konfiguraci registrovaných zařízení. Aby uživatel mohl konfigurovat vybrané zařízení, musí být pro toto zařízení autorizován.

Odstranění uživatele

Uživatelský účet může oprávněný uživatel bez návratu odstranit, svůj vlastní účet si uživatel odstranit nemůže.

4.3.2 Správa ROS zařízení

Konfiguraci ROS pomocí IS může provádět pouze přihlášený uživatel, který je oprávněn k provedení dané operace (má povolen přístup do požadované konfigurační části) a zároveň má na konkrétní zařízení oprávnění.

Registrace nového zařízení do systému

Každé zařízení s ROS je nutné v systému nejdříve zaregistrovat. K registraci je nutná přístupová IP adresa a přihlašovací údaje. Přidat nové zařízení může jen uživatel s oprávněním pro přidávání nových zařízení. Uživatel, který zařízení zaregistroval, automaticky získá oprávnění k jeho konfiguraci. Přístupová IP adresa je zaznamenána.

Evidence připojených zařízení

Když je zařízení zaregistrované v systému, je o něm možné zobrazit základní informace, jako je verze ROS či hardwarová konfigurace. V IS jsou uchovány údaje o všech existujících rozhraních (IP adresy, aktivní DHCP server, SSID a mnoho dalších). V rámci evidence je umožněn rychlý náhled na aktuální stav zařízení.

Konfigurace zařízení

U všech registrovaných RB je možné nastavit alespoň jméno zařízení. Informace o konfiguraci ROS se ukládají i mimo RB. K zařízení je možno přidat krátký popis.

Konfigurace rozhraní

IP adresy rozhraní nelze konfigurovat, nevhodnou manipulací by mohlo dojít k znepřístupnění některých segmentů sítě. Stejně tak není dovolená konfigurace směrovacích protokolů. Povolená je naopak konfigurace DHCP serveru. U bezdrátových rozhraní se dá nastavit komunikační frekvence a SSID.

Možnost obnovy ROS po havárii

Při poškození RB je možné přenést uloženou konfiguraci do nového zařízení. Zálohy jsou vytvářeny uživatelem.

Odstranění zařízení ze systému

Zařízení je možné odstranit z evidence systému. Později ho lze opět přidat.

4.3.3 Správa připojených klientů

Správa připojení klientů je klíčovou funkcí tohoto systému. Je ale nutné mít na paměti, že ROS nedisponuje žádnou triviální funkcí, která umožňuje blokování fyzické MAC adresy u připojení pomocí kabelu. Proto zde uvedené funkce nebudou schopny dostatečně omezit přístup přes kabelové připojení.

U všech níže uvedených funkcí je vždy důležité, aby uživatel, který chce prohlížet či upravovat připojení, měl odpovídající oprávnění pro správu RB, ke kterému je připojení realizované. Dále musí mít autorizaci pro provedení daného úkonu v systému.

Registrace připojení

Před tím, než se klient může připojit do sítě, je nutné zaregistrovat ho v systému. K registraci je třeba MAC adresa jeho síťové karty, jméno a kontaktní elektronická adresa. Po registraci je klient evidován v systému a čeká na aktivaci.

Evidence připojených klientů

Je možné zobrazit všechny registrované připojení, ať už aktuálně aktivované nebo deaktivované. U všech klientů lze zobrazit jejich MAC adresu, kontaktní email a informaci o aktuálním stavu připojení. U aktivovaného klienta je možné zjistit jeho IP adresu a další informace o připojení. V případě změny lze uložené informace o klientovi upravit.

Aktivace připojení

Chce-li se klient připojit do sítě, musí ho nejprve některý uživatel v systému aktivovat. Při aktivování je klientovi přidělena IP adresa. Tato adresa je spolu s MAC adresou zapsána přímo do konfigurace ROS.

Deaktivace připojení

Při deaktivaci připojení je klientovi zablokován přístup do sítě. Jeho IP adresa je odebrána a může být přidělena jinému klientovi. V tomto okamžiku může být klient připojen do části sítě. V případě nové aktivace ve stejné síti může být klientovi přidělena jiná IP adresa.

Omezení rychlosti

U všech připojených klientů lze nastavit maximální rychlost připojení. Omezení rychlosti ovlivní rychlost příjmu a odesílání dat, případně obojí.

Odstranění připojení ze systému

Nevyhovující či již neexistující připojení lze z evidence kdykoliv odstranit. Jejich pozdější přidání je opět možné.

4.3.4 Zobrazení statistik

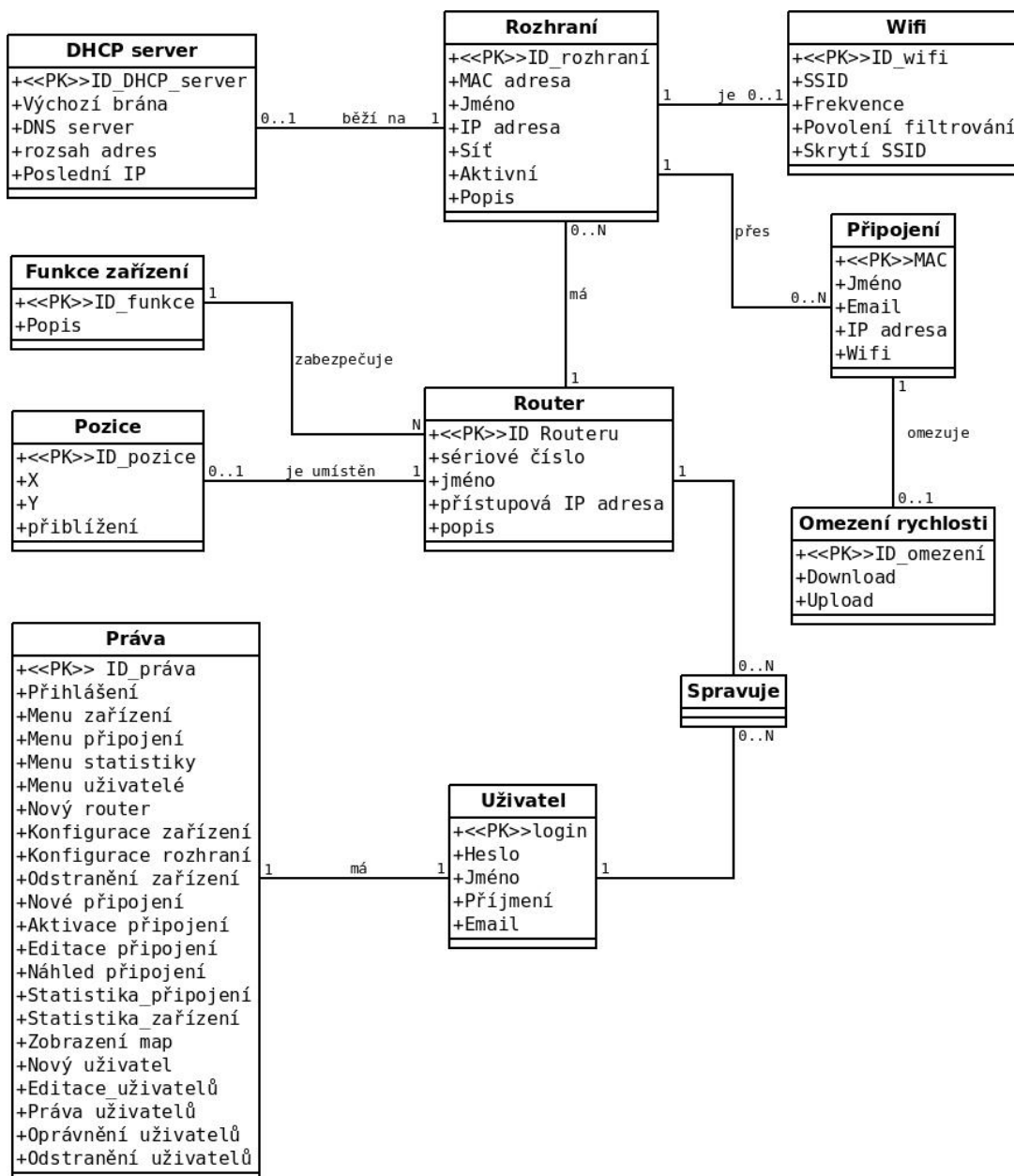
ROS umožňuje vytváření jednoduchých statistik. Navrhovaný systém musí tyto statistiky v ROS nejdříve povolit a následně také zobrazit v podobě grafu. Je možné zobrazit všechny dostupné grafy uvedené v 2.6. Přihlášený uživatel má přístup ke statistikám jen tehdy, pokud má vyhovující oprávnění.

4.3.5 Propojení s mapovým podkladem

Fyzická pozice každého registrovaného RB může být zaznamenána na mapě a poté na ní zobrazena. V jedné mapě lze zobrazit jak jediné zařízení tak i všechny dostupné. Použití dostupných mapových podkladů je vázáno na připojení k internetu, Proto je nutné, aby počítač, na kterém bude IS nasazen, měl přístup do internetu, v opačném případě nelze mapy použít.

4.4 E-R diagram

Na základě výše uvedené množiny funkcí je sestaven tento E-R diagram, který slouží jako základ pro vytvoření databázového modelu.



Obrázek 4.1: E-R diagram.

Kapitola 5

Implementace navrženého systému

V této kapitole projdeme stěžejní část této práce, kterou je implementace jednotlivých částí už navrženého systému. Úvodní podkapitola je věnována výběru programovacích jazyků a vysvětlení některých, zde užitých pojmů z oblasti objektově orientovaného programování. Dále popíšeme rozložení adresářů se zdrojovými texty, zmíníme navrženou databázi a třídy, které s databází komunikují. Ukážeme si, jak systém generuje obsah stránek, jakým způsobem je zajištěna konfigurace ROS a zobrazení map. V závěru projdeme implementaci klíčových částí systému. Kapitola je doplněna i o základní metriky zdrojového kódu.

5.1 Zvolený programovací jazyk

Kritéria pro volbu programovacího jazyka jsou již předem daná. Podle zadání práce se pro tvorbu webu musí použít jazyk HTML ve spolupráci s CSS a pro skriptování jazyk PHP. Data se ukládají do MySQL databáze. Na klientské straně je použit JavaScript.

Jazyk HTML je standardním jazykem pro tvorbu hypertextových dokumentů. Tento jazyk tvoří základ všech internetových stránek umístěných na internetu [4]. HTML ale není dostačující pro tvorbu grafického uživatelského rozhraní, proto se musí spolu s HTML použít kaskádové styly (CSS). V HTML je tak nadefinována jen struktura stránky a její obsah. Vzhled stránky je definován odděleně [3].

PHP je jazyk, který spouští skripty na straně serveru a klientovi předává už interpretovaný výsledek. Ke zdrojovému kódu se tak dostane jen jeho tvůrce. Při implementaci zadaného IS je využito principů objektově orientovaného programování (OOP), které jazyk PHP5 již plně podporuje. Oproti procedurálnímu programování, kde jsou funkce a data od sebe navzájem odděleny, jsou v OOP data a funkce, které nad těmito daty operují, navzájem svázány do objektů. Objekty obsahují data (nazývané atributy nebo vlastnosti) a funkce operující nad těmito daty (nazývané metody). Objekt je definován třídou, jejíž je instancí. Vytvořením instance se vytvoří nový objekt inicializující všechny atributy a zároveň je volán konstruktor — funkce provádějící prvotní nastavení objektu [15]. Všechny používané PHP funkce byly vyhledány v manuálu pro PHP [2].

5.2 Adresářové schéma

V kořenovém adresáři je uložen soubor `index.php`, který slouží jako vstupní brána do systému. Ve stejném adresáři se nachází i soubor s kaskádovými styly `styl.css` a konfigurační

soubor `general.php`. Odkazy na oba soubory jsou vkládány do každé stránky. V konfiguračním souboru jsou uloženy přístupové údaje potřebné pro připojení do databáze, síťový port MikroTik API, titulek stránek a další drobná nastavení. Mimo uvedené konstanty obsahuje soubor funkci `__autoload`, která sama vyhledá umístění požadované třídy. Na začátek každé stránky pak stačí vložit následující příkaz: `require_once '../general.php';`

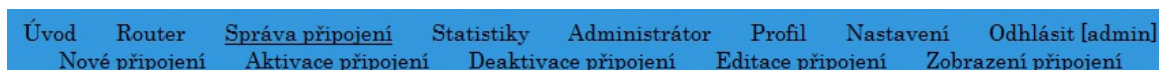
Jednotlivé stránky se nachází v adresáři `page`. Každá stránka je pojmenována podle názvu sekce, například `connect_activation`. Stejné označení je užíváno i uvnitř skriptů. Názvy všech sekcí:

- připojení (`connect`)
- uživatel (`netadmin`)
- konfigurace (`router`)
- statistiky (`stat`)
- profil

V adresáři `lib/class` jsou uloženy všechny existující třídy. Pro každou třídu je vyhrazen samostatný soubor. Pro uložení JavaScriptu slouží adresář `lib/script`.

5.3 Generování obsahu HTML stránek

Generování HTML dokumentu zajišťuje třída `HTML`. Třída je naimplementována pomocí návrhového vzoru Singleton (popsán v [15]). Při generování každé stránky se vždy vytvoří pouze jediná instance třídy. O generování začátku HTML stránky se stará metoda `startHTML`, jako parametry se předávají: název stránky, hloubka zanoření stránky v adresáři (typicky hodnoty 0 nebo 1), název skriptu, který se vykoná po obnově stránky a pokud je do stránky vložena mapa, je nutné vložit do posledního parametru hodnotu `true`. Při opakovaném volání této třídy se vyvolá výjimka. Pro vygenerování konce dokumentu slouží metoda `endHTML`. I zde je ošetřeno několikanásobné volání, navíc je zabráněno zavolání metody před vygenerováním začátku. Třída umožňuje vygenerovat i hlavičku stránky. Poslední důležitá metoda implementovaná v této třídě je metoda `generate_menu`. Metoda generuje obslužné menu systému na základě oprávnění přihlášeného uživatele. Oprávnění jsou načtena při každém zavolání této metody. Pokud nastanou změny v uživatelském oprávnění, jsou v systému promítnuty ihned po opětovném volání metody. Obsah generovaného menu se rozlišuje pomocí parametru. Pokud je hodnota stejnojmenně pojmenovaného sloupce tabulky `rights` v databázi nastavena na logickou 1, je sekce přístupná.



Úvod	Router	<u>Správa připojení</u>	Statistiky	Administrátor	Profil	Nastavení	Odhlásit [admin]
Nové připojení	Aktivace připojení	Deaktivace připojení	Editace připojení	Zobrazení připojení			

Obrázek 5.1: Základní menu.

Pro opakující se funkce je vyhrazena třída `page_function`. Všechny metody jsou statické. Často se jedná o metody, které je třeba zavolat ještě před vygenerováním HTML hlavičky, nejčastěji z důvodu přesměrování pomocí PHP funkce `header`, dále jsou zde umístěny metody, které vytváří vhodně naplněné seznamy pro výběr hodnot (tzv. combo boxy).

Interakce s uživatelem probíhá pomocí formulářů. Vyplněný formulář je ve všech případech posílán skrytou http metodou POST. Zpracování provádí vždy ta stránka, ze které je formulář s daty odesílán. V PHP je tomuto účelu vyhrazena konstanta `PHP_SELF`. Během generování takovéto stránky, se vždy kontroluje přes PHP funkci `isset`, zda není odeslán některý z formulářů umístěných na stránce, pokud ano, je na vzniklou událost patřičně zareagováno. Každý objekt implementuje metodu `generate`, která podle typu objektu generuje obsah stránky.

5.4 Databáze

Všechna data systému se uchovávají v relační databázi MySQL. Jedná se o volně dostupnou open-source databázi, která se vyznačuje svojí rychlostí a jednoduchostí [14]. Jazyk PHP obsahuje širokou škálu funkcí pro práci s touto databází.

5.4.1 Schéma relační databáze

Z E-R diagramu uvedeném na obrázku 4.1 vznikne po transformaci schéma zobrazené na obrázku 5.2 (přebráno z použité databáze, a proto ty anglické názvy).

5.4.2 Přístup do databáze

Komunikaci s databází obstarávají tři třídy, dvě z toho se starají přímo o komunikaci a jedna třída uchovává nejčastěji používané dotazy. Dotazy jsou jinak vždy definovány na místě jejich použití. První dvě třídy jsou z části opět přebrány z [15].

Třída `mysql`

Účelem této třídy je napojit se na databázi. Její konstruktor inicializuje uživatelské jméno, heslo, adresu MySQL hostitele a jméno databáze. Po vytvoření její instance je dostupná metoda `prepare`. Metoda vrací nový objekt typu `statement`, který specifikuje dotaz zadaný pomocí parametru.

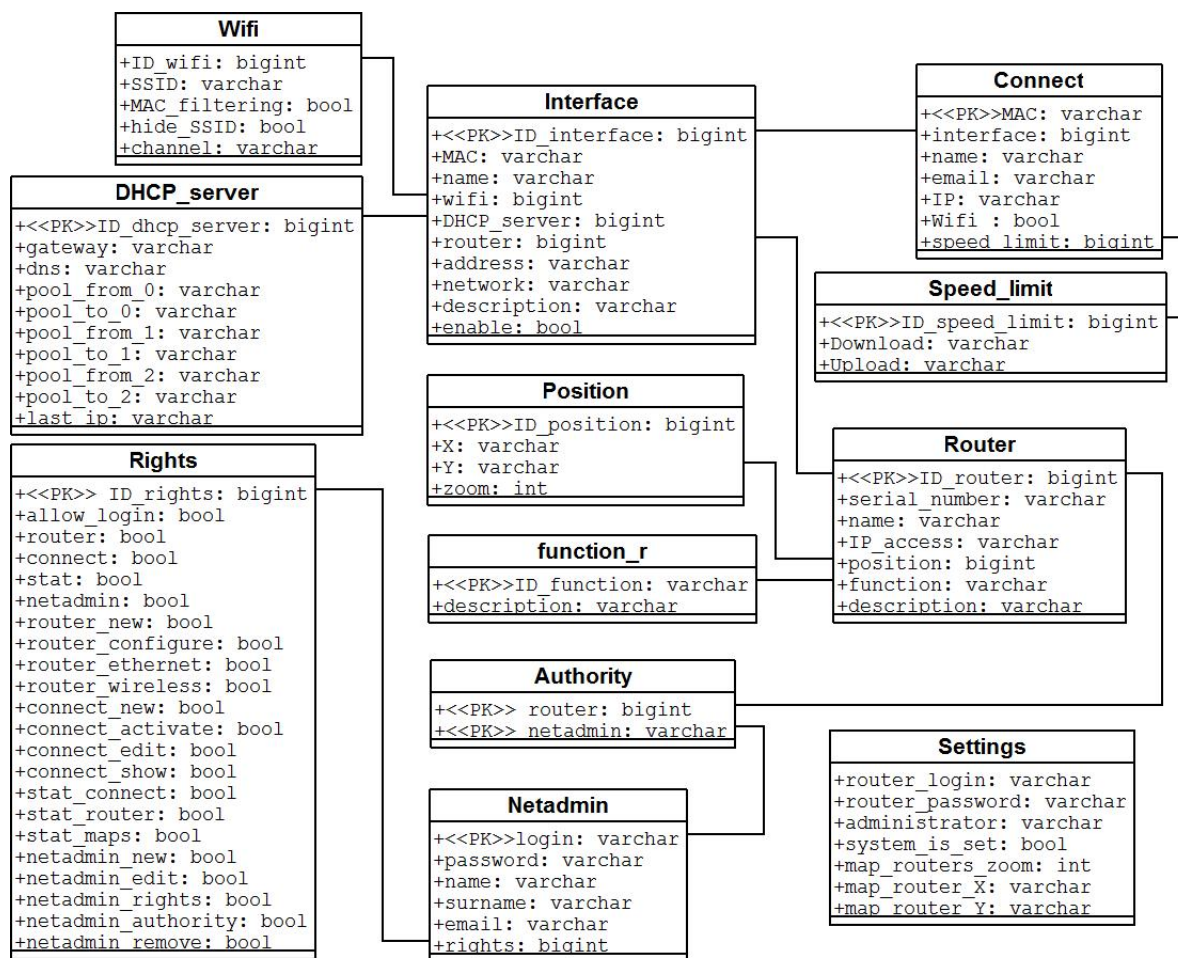
Třída `statement`

Třída zaobaluje výsledek provedení dotazu a implementuje metody, které slouží k získání výsledných dat. Nejprve je nutné zavolat metodu `execute`, která dotaz vykoná. Až následně jsou dostupné funkce pro samotné získání dat. Dostupná je metoda `fetch_assoc`, která vrací jeden načtený řádek výsledku nebo například metoda `fetch_num_rows` vracející počet řádků získaného výsledku.

Třída `query`

Nejčastěji používané dotazy jsou umístěné do této třídy. Přístup k jednotlivým dotazům, které jsou uloženy v asociativním poli, je pomocí statických metod. K hodnotám pole se přistupuje přes metodu `get_query`. Navíc jsou ve třídě umístěny i metody, které vracejí už instanci typu `statement`. Každý dotaz je definován tak, aby mohl být použitý s libovolnými parametry. Parametry se dosazují pomocí PHP funkce `sprintf`. Pro úplnost je uveden příklad takto definovaného dotazu:

```
UPDATE connect SET interface = %d, IP = "%s" WHERE MAC = "%s"
```



Obrázek 5.2: Schéma relační databáze.

5.5 Konfigurace ROS

Veškerá komunikace mezi informačním systémem a ROS je založena na třídě popsané v kapitole 3. Pro potřeby systému je tato třída přeimplementována na abstraktní, a proto třída, která chce konfigurovat ROS musí být potomkem této abstraktní třídy. V zdrojovém kódu se jedná konkrétně o třídu `routeros_api`, v této třídě je navíc definovaná i abstraktní metoda `generate`. Každá odvozená třída musí metodu implementovat. Jak bylo zmíněno výše, jejím primárním účelem je generování HTML kódu, který je specifický pro každou odvozenou třídu.

Konkrétní příkazy zasílané ROS jsou obsaženy v metodách implementovaných přímo ve třídě `routeros_api`. Pro přehlednost jsou všechny metody, které obsahují konfigurační příkazy umístěny právě v této jediné třídě a začínají zkratkou `API`. Tělo metody je vždy tvořeno kombinací příkazů `read`, `write` a `comm`.

V podstatě se opakují tři základní způsoby implementace. Při prvním způsobu je zaslána skupina příkazu (i jeden příkaz) pomocí `write` a následné čtení odpovědi pomocí `read`. Příkladem je metoda `API_wireless_info`, která vrátí informace o všech dostupných rozhraní umístěných na RB. Druhý způsob je vyhledání a filtrování pomocí `comm`. Zde lze jako příklad

uvést metodu `API_get_interface_ip` vracející IP adresu rozhraní, jehož jméno je zadáno parametrem. Při posledním způsobu se nejdříve podle zvolených parametrů metodou `comm` vyhledá identifikační číslo (.id). Následně se pracuje s výsledky pomocí dalších metod `comm` či metod `write`, na závěr se opět volá metoda `read`. Jako příklad může být uvedena metoda `API_set_interface_enable`, sloužící k zapnutí/vypnutí zvoleného rozhraní.

5.6 Zobrazení na mapě

Zobrazení map zajišťuje aplikace **Google Maps**, která je pro nekomerční užití zdarma k dispozici. Mapy jsou do jednotlivých stránek vkládány pomocí **Google Maps API**. K dispozici je varianta pro JavaScript a pro Flash. Zde popisovaný IS používá Google Maps Javascript API verze 3 [6].

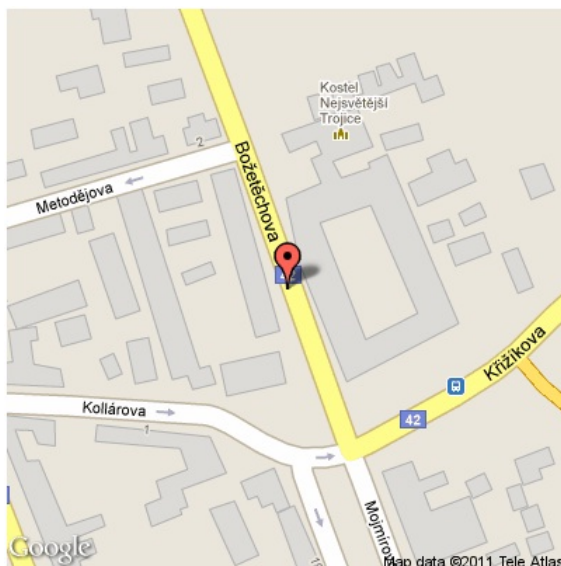
Google maps jsem vybral kvůli neomezenému počtu zobrazení dynamických map a perfektní dokumentaci s velkým počtem názorných příkladů. Nespornou výhodou je, že systém nevyžaduje použití klíče, jak tomu je u jiných mapových systémů [5].

5.6.1 Statické mapy

Google nabízí službu, která umožňuje vkládat i mapy do stránek i bez použití JavaScriptu, jedná se o tzv. statické mapy [7]. Mapa do stránky je vložena přes HTML tag ``, kde jako zdroj obrázku je použit speciální odkaz. Tvar odkazu:

<http://maps.google.com/maps/api/staticmap?parameters>

Statické mapy jsou limitovány počtem zobrazených map. Lze zobrazit 1000 různých map v průběhu 24 hodin. Pro popisovaný informační systém to ale limitující určitě není, navíc stejná mapa může být zobrazena několikrát. V systému jsou statické mapy použity na stránce, která zobrazuje s bližšími informacemi o RB.



Obrázek 5.3: Příklad statické mapy.

5.6.2 Použití v systému

Do hlavičky každé stránky zobrazující mapu je nutné vložit tento malý skript:

```
<script  
type="text/javascript"src="http://maps.google.com/maps/api/js?sensor=true">  
</script>
```

Díky tomu je možno používat Google Maps API. Skript je do stránky vložen v metodě `startHTML` (poslední parametr nastaven na `true`). Spolu s ním je do stránky umístěn odkaz, na zdrojový kód, který zajišťuje požadovanou obsluhu. Jedná se o skript `maps.js` v adresáři `lib/script`. Tento skript obsahuje čtyři základní funkce, které jsou spouštěny přímo ze stránky. Tyto funkce slouží k vytvoření nebo editaci pozice, dále pak k zobrazení jediného nebo všech zařízení.

Každá z posledně jmenovaných funkcí nastaví atributy a odešle `XMLHttpRequest` (funkce na základě požadavku zajišťuje výměnu dat mezi klientem a serverem bez znovunačtení stránky [16]). Požadavek je adresován skriptu `load_map_info`, který generuje XML soubor. Přes `http` metodu `GET` mu je předáno identifikační číslo (ID) zařízení, na jehož základě skript vrátí údaje o konkrétním zařízení. Při jeho neuvedení jsou vráceny údaje o všech zařízeních, která může uživatel konfigurovat. Pokud je ID rovno nule, není žádné zařízení vráceno. V tomto případě je navrženo pouze výchozí nastavení mapy (vrací se vždy).

Na základě odpovědi je vytvořena mapa (objekt `google.maps.Map`). Střed mapy tvoří výchozí uložená pozice nebo přímo pozice zobrazeného zařízení. Až poté následuje zpracování pozic jednotlivých zařízení a jejich zobrazení v podobě značek (`google.maps.Marker`). Ke značce je vždy navázáno provedení události. V tomto případě se po kliknutí na značku zobrazí „vyskakovací“ popisek (objekt `google.maps.InfoWindow`) nebo je značka odebrána. Pokud uživatel zadává novou pozici zařízení, je pozice značky včetně aktuálního přiblížení zobrazena v textových polích, které jsou součástí formuláře. Pomocí formuláře jsou získané údaje uloženy do databáze. Popisovaný princip je použit na stránkách `router_new`, `router_configure` a `settings`. Zobrazená zařízení jsou graficky vybarvena podle aktuálního stavu (dostupná zařízení zeleně, nedostupná červeně). Pomocí takovéto mapy je možné získat základní přehled o stavu sítě, v systému je takovou mapu možno nalézt na stránce `stat_maps`.

5.7 Implementace funkcí systému

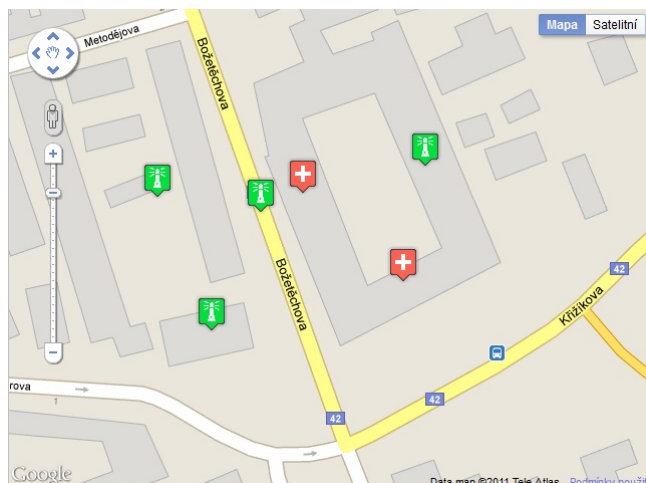
Popis implementace uváděných funkcí je rozdělen na dvě části. První část ukazuje implementaci uživatelské obsluhy a část druhá popisuje konkrétní úkony provádění při interakci s dostupnými RB.

5.7.1 Uživatelská obsluha

Kompletní správu uživatelů IS zajišťuje třída `web_netadmin_user` a pokud není uvedeno jinak, jsou všechny zde uvedené metody naimplementovány v této třídě.

Přihlášení do systému

Na hlavní stránce (skript `index.php`) je vygenerován formulář, díky kterému se uživatel pomocí uživatelského jména a hesla může autentizovat do systému. Údaje získané z formuláře se předají metodě `login`, která se v databázi pokusí vyhledat požadovanou kombinaci



Obrázek 5.4: Dostupná a nedostupná zařízení zobrazená na mapě.

uživatelského jména a hesla. V případě vyhledání právě jednoho záznamu se jedná o úspěšné přihlášení. Vytvoří se dvě superglobální proměnné (tzv. session) uchovávající přihlašovací jméno uživatele a přihlašovací jméno hlavního administrátora. Následně je uživatel pomocí vestavěné PHP funkce `header` přesměrován na hlavní stránku systému. V případě neúspěšného přihlášení je uživatel o dané skutečnosti informován a zadává přihlašovací údaje znovu.

Odhlášení ze systému

Uživatel se může odhlásit ze systému po kliknutí na tlačítko odhlásit, které je umístěno v hlavním menu. Odhlášení spočívá v přesměrování na úvodní stránku, na kterou je odeslán pomocí http metody `get` stav „logout“. Po detekci tohoto stavu je zavolána metoda `logout`, která zajistí odstranění superglobálních proměnných. Druhou možností, jak může být uživatel ze systému odhlášen, je jeho delší neaktivita.

Tvorba nového uživatele

Nový uživatel je vytvořen metodou `register_new`. Přes formulář pro vytvoření uživatele, umístěný na stránce `netadmin_new`, je získáno uživatelské jméno, heslo, heslo pro kontrolu, jméno, příjmení a email. Všechny obdržené hodnoty jsou přes parametry předány uvedené metodě a až zde se kontroluje vyplnění všech položek, korektnost emailu a totožnost zadaných hesel. Pokud není detekován žádný problém, je uživatel uložen do databáze. Spolu s uživatelem je v databázi vytvořen záznam s jeho oprávněním. Vzájemný vztah mezi záznamy je 1 : 1. Záznamy jsou od sebe odděleny za účelem vyšší úrovně abstrakce.

Oprávnění uživatelů

Systém rozlišuje tři úrovně oprávnění. Na první úrovni se kontroluje, zda je uživatel do systému autentizován, konkrétně se kontroluje existence superglobální proměnné `login`. Druhá úroveň se týká omezení přístupu do jednotlivých částí systému. Princip byl z části naznačen při popisu generování menu v 5.3. Dochází ke dvojité kontrole, kdy ještě před vygenerováním menu je zkontrolováno, jestli má uživatel povolen přístup do dané sekce. Pokud

ne, je uživatel automaticky přesměrován na stránku `no_access`. K nastavení této úrovně oprávnění slouží formulář na stránce `netadmin_rights` (Obr. 5.5). Při zpracování odeslaného formuláře se z odeslaných dat vytvoří pole, které je předáno metodě `set_rights`. Za zmínku stojí PHP funkce `vsprintf`, která předané pole dosadí do aktualizujícího dotazu.

Login:	xlesti00	
Přihlášení:	ANO ▾	
Správa routerů	NE ▾	
Přidat nový router:	NE ▾	Konfigurace routerů: NE ▾
Konfigurace ethernet:	NE ▾	Konfigurace wireless: NE ▾
Odstranění routeru:	NE ▾	
Připojení uživatelů	NE ▾	
Nové připojení:	NE ▾	Editace: NE ▾
Aktivace připojení:	NE ▾	Deaktivace připojení: NE ▾
Zobrazení připojení:	NE ▾	
Přístup do statistik	NE ▾	
Statistiky routerů:	NE ▾	
Statistiky uživatelů:	NE ▾	Mapy: NE ▾
Administrátor	NE ▾	
Přidat nového administrátora:	NE ▾	Editace: NE ▾
Odstranění:	NE ▾	
Nastavení práv:	NE ▾	Autorizace: NE ▾

Obrázek 5.5: Formulář pro nastavení oprávnění.

Třetí úroveň zajišťuje přístup k jednotlivým zařízením. ROS totiž neumožňuje specifikovat autorizaci jen pro vybranou množinu funkcí, proto se informační systém k jednotlivým zařízením připojuje přes přihlašovací účet s plným oprávněním a až v systému jsou jednotlivé funkce filtrovány. Přístup k zařízením lze nastavit pomocí skriptu `netadmin_authority`. Uživatel v seznamu všech evidovaných zařízení vybere to, které chce konfigurovat a po potvrzení volby mu je nabídnuto pole zaškrtačacích tlačítek (check box) s přihlašujícími jmény všech existujících uživatelů. Pole je předvyplněno podle existujících oprávnění. Při odeslání formuláře se zaškrtačacími položkami jsou odeslány jen ty zaškrtnuté. Uložení autorizace pak probíhá tak, že jsou z databáze postupně načítáni všichni uživatelé a porovnává se, zda byl aktuálně načtený uživatel ve formuláři odeslán. V případě shody je zavolána metoda `set_authority`, která přidá do tabulky autority záznam svazující uživatele se zařízením. Pokud uživatel ve formuláři vybrán nebyl, následuje volání metody `remove_authority`, které odstraňuje kombinaci uživatelského jména a zařízení z tabulky autority. Omezení oprávnění se v systému projeví tak, že uživateli nejsou vůbec nabídnuta zařízení, ke kterým nemá přístup. Pro zajištění tohoto omezení je nutné vhodně rozšířit MySQL dotazy, které získávají informace o zařízeních.

Editace základních údajů

Každý uživatel může upravit veškeré o sobě uváděné informace ve svém uživatelském profilu. Po jeho zobrazení (menu `profil`) je nabídnut předvyplněný formulář. Po jeho odeslání je volána metoda `edit`. V rámci ní jsou zkontrolovány vyplněné údaje a až poté následuje případná aktualizace databáze. Na stejné stránce je umístěno i jméno, příjmení a email

pro kontaktování ostatních uživatelů. Editace základních údajů je proveditelná i z menu zajišťující správu uživatelů (stránka `netadmin_edit`). Nejprve je nutné zvolit uživatele z výběrového seznamu a až poté je zpřístupněn stejný formulář jako v menu profil.

Odstranění uživatele

V PHP skriptu `netadmin_remove` jen generován výběrový seznam obsahující všechny uživatele, které lze smazat. Seznam automaticky nenabízí uživatelské jméno přihlášeného uživatele a hlavního administrátora. Až po potvrzení vybraného uživatele je volána metoda `remove`.

Nastavení systému

Ohledně prvního přihlášení je administrátor vyzván, aby provedl úvodní nastavení, místo na obvyklou stránku je proto přesměrován na stránku `setting`, na které administrátor musí do formuláře zadat uživatelské jméno a heslo. Jedná se o univerzální přihlašovací údaje, na jejichž základě je umožněn přístup ke všem registrovaným zařízením, po uložení už tento účet není možné měnit. Bez vyplnění tohoto formuláře systém nemůže pracovat (práce umožněna ani není, administrátor nemá doposud žádné oprávnění). Tento účet vytváří systém automaticky vždy při registraci zařízení. Veškerá systémová nastavení uvedená na této stránce jsou zpracovávána ve třídě `system_settings`. Uložení dat načtených z formuláře zajišťuje metoda `save`. Až v rámci této metody je administrátorovi přiděleno plné oprávnění pro správu systému. Nastavení systému se týká i mapových podkladů. Administrátor může nastavit výchozí pozici map (startovací pozice při přidání nového zařízení a náhledu na všechna zařízení).

5.7.2 Interakce s RB

Základní práci (registraci, editaci a nastavení rozhraní) zajišťuje třída `router_setting`. Další funkčnost (aktivace klientů, zobrazení statistik, zálohování a další) zajišťují vždy třídy k tomu určené. Je vhodné připomenout, že objekt, který komunikuje s ROS přes MikroTik API je potomkem abstraktní třídy `rotueros_api`. Ke všem registrovaným zařízením se systém připojuje pomocí univerzálního přihlašovacího účtu.

Registrace nového zařízení

V menu router, na stránce `router_new` je umístěn jednoduchý formulář, do kterého je třeba zadat IP adresu nově registrovaného zařízení a údaje nutné pro přihlášení (uživatelské jméno, případně heslo). Je nutné, aby uživatelský účet měl plná ROS oprávnění. Připojené zařízení musí mít samozřejmě povolené přihlášení pomocí API (příkaz [2.5.5](#)).

Po odeslání formuláře, je otestováno, zda není stejná IP adresa již registrovaná a zda má systém k zadanému zařízení vůbec přístup. Pokud ano, je ze zařízení získáno sériové číslo RB, které se následně zkouší vyhledat v databázi. V případě, že se sériové číslo v databázi již nachází, je o tom uživatel informován a registrace končí.

Na stránce je zatím vygenerován formulář obsahující mimo jiné i základní informace o IP adresách. Ve formuláři lze nezávazně vybrat funkci, kterou zařízení v síti plní a přidat krátký popis. Je možné zvolit i umístění zařízení na mapě.

Pro úspěšnou registraci zařízení je nutné odeslat i tento formulář. Odeslaná data zpracovává metoda `add_new`. Metoda nemá žádné parametry a data, se kterými pracuje, získává

pomocí POST. Důvodem je předem neznámý počet rozhraní RB. Metoda nejdříve kontroluje jestli je zadáno jméno zařízení, pod kterým bude v IS evidováno. Následuje vytvoření univerzálního přihlašovacího účtu do ROS. V této části je zařízení uloženo do databáze (zatím jen zařízení bez přiřazených rozhraní).

Z databáze je načten identifikátor posledního vkládaného údaje (aktuálně registrované zařízení). Identifikátor je spolu se jménem přihlášeného uživatele vložen do tabulky `authority` (vytvořena instance třídy `web_netadmin_user` a metoda `set_authority`), spolu s přihlášeným uživatelem systém spáruje zařízení i s hlavním administrátorem. Identifikátor je předán také metodě `save_interface_information`, která ukládá všechna existující rozhraní. Metoda zajišťuje rozsáhlou činnost, později proto bude popsána podrobněji.

Na závěr systém nakonfiguruje ROS tak, aby zaznamenával statistiky. Dioda umístěná na zařízení bude po dobu 5 vteřin blikat (znamení úspěšné registrace).

Editace zařízení

Systém dovoluje editovat jen dostupná zařízení. Nabízena jsou formou výběrového seznamu, který generuje statická metoda `create_online_router_combobox` umístěná ve třídě `page_function`. Metoda se postupně připojuje k evidovaným zařízením a v případě úspěšného připojení je zařízení zobrazeno v seznamu. Pro vybrané zařízení je vygenerován podobný formulář jako při vlastní registraci (není generován formulář pro editaci rozhraní). Stránka je rozšířena i o možnost nastavit datum a čas zařízení ve zvláštním formuláři. Oba formuláře mají oddělená spouštěcí tlačítka. Pro potvrzení změn v nastavení zařízení je volána metoda `edit_configure`. Datum a čas nastavuje metoda `edit_time`. Uložení je také zde vizuálně potvrzeno blikáním diody.

Editace rozhraní

Každé zařízení je v databázi zaregistrované včetně všech informací o rozhraní a stejně jako ROS, rozlišuje systém rozhraní na bezdrátové a drátové (ethernet). Pro editace každého typu je vyhrazeno zvláštní menu. Pro bezdrátové je to menu `router_wireless`, pro drátové `router_ethernet`. Informace o obou rozhraních jsou uchovány v systémové databázi. Editační formuláře však načítají informace vždy přímo ze zařízení.

Formulář pro editaci rozhraní generuje třída `router_interface_list`. V rámci této třídy je rozlišeno o jaký druh rozhraní se jedná. U obou rozhraní je možné nahlédnout na informace týkající se IP adres. Nepřístupové rozhraní lze vypnout. Umožněna je aktivace DHCP serveru. U DHCP serveru lze nastavit přidělovanou výchozí bránu a DNS server. Dále je možné specifikovat až tři rozsahy, ze kterých budou přidělovány klientům adresy. DHCP server musí být povolen na všech rozhraních, na kterých se budou připojovat klienti. U bezdrátového rozhraní může uživatel navíc upravit název bezdrátové sítě (SSID), skrýt SSID, vypnout MAC autentizaci pro přístup uživatelů a nastavit kanál, na kterém rozhraní vysílá. Zabezpečit bezdrátovou síť pomocí klíče systém neumožňuje.

I zde platí, že lze konfigurovat jen ta zařízení, která jsou aktuálně přístupná (v tomto případě jsou nabízena jen dostupná rozhraní). Metoda `save_interface_information` zajišťuje uložení rozhraní. Jelikož každý typ RB může obsahovat různý počet rozhraní, metodě je předán pomocí parametru seznam dostupných rozhraní načtený z konfigurovaného RB. Další parametr specifikuje typ rozhraní. Přímou v metodě jsou v cyklu procházena jednotlivá rozhraní a kombinací názvu rozhraní s předem stanoveným jménem odesílaného prvku, jsou zpracovávány údaje zaslané pomocí POST. Stejným způsobem jsou generovány prvky ve formuláři. Příklad získání dat: `$_POST[$iname."_active"]`, kde hodnota proměnné `iname`

je jméno procházeného rozhraní. V rámci tohoto cyklu je do databáze ukládáno a do konfigurace ROS zapsáno i případné nastavení DHCP serveru a rozšířené nastavení, které vyžadují bezdrátová rozhraní. Při úspěšném uložení dioda na zařízení krátce zabliká.

Odstranění zařízení

Stránka `router_remove` obsahuje výběrový seznam, který nabízí všechny registrované zařízení. Obsah stránky a pozdější odstranění ze systému zajišťují metody z třídy `remove_router`. Systém odstraní záznam o zařízení z databáze. Odstraněny jsou i záznamy, které jsou se zařízením spojené (rozhraní, pozice a další). Připojení klientů vázané na toto zařízení je nastaveno na neaktivní, v zařízení však zůstane povolené.

Obnova zařízení

Pokud dojde k poruše registrovaného zařízení, je nutné ho ze systému odstranit a zaregistrovat nové zařízení a to znovu nakonfigurovat. Řešení je to funkční ale nepraktické, proto systém nabízí možnost přenesení konfigurace předchozího RB do nového RB bez odstranění ze systému. Zálohy konfigurací musí uživatel manuálně vytvářet. Funkce pro zálohování a obnovu zařízení jsou definovány ve třídě `router_backup`. Důležité jsou metody `backup` a `restore`.

Uložit konfiguraci je možné na stránce `router_configure`. Po stisku určeného tlačítka je zavolána metoda `backup`, v rámci ní je vyhledáno sériové číslo RB, které je následně použito pro pojmenování zálohy (vždy je uchovávána jen jedna aktuální záloha). Vytvoření zálohy provede ROS funkce `export`. Zálohu ROS uloží do paměti zařízení a odtud je pomocí FTP přenesena do složky `page/backup` umístěné na serveru.

Obnova zařízení probíhá pomocí registrujícího formuláře (stránka `router_new`), ve formuláři uživatel vyplní údaje potřebné k připojení a volbu „Obnova po poruše“ nastaví na hodnotu „ano“. Poté formulář může odeslat. V případě úspěšného připojení k zařízení systém detekuje, že se jedná o obnovu. Uživatel nyní musí potvrdit provedení obnovy. Po odeslání potvrzujícího tlačítka je zavolána metoda `restore`.

Nyní následuje opačný postup oproti získání zálohy. Systém vyhledá v databázi sériové číslo zařízení (pomocí zadané IP adresy). Následně aplikace vyhledá, zda záloha s registrovaným číslem existuje, pokud ne, je o tom uživatel informován a obnova končí neúspěšně. V opačném případě, je opět pomocí FTP záloha přenesena do paměti RB. V ROS je poté spuštěn příkaz `import`. Příkaz `import` nemusí být vždy úspěšně dokončen. Při obnovení může dojít ke kolizi stejných názvů. V takovém případě je nutné uvést zařízení do továrního nastavení, nastavit jen přístupovou IP adresy a celý postup zopakovat. Při úspěšné obnově zařízení krátce zabliká.

Zobrazení registrovaných zařízení

Nabídka `Router` dává uživateli možnost získat přehled o stavu jeho zařízení. Zobrazené záznamy obsahují informace o názvu zařízení, funkci zařízení a přístupovou IP adresu. Záznamy jsou navíc podle dostupnosti vizuálně rozlišené. Každý záznam obsahuje odkaz pro bližší informace. Záznamy jsou generovány pomocí třídy `preview`. Konstruktor tohoto objektu inicializuje pouze přístupové uživatelské jméno a heslo. Pro vygenerování záznamu metodou `generate`, je nutné jako parametr předat načtený řádek z databáze (přístupová IP adresa, jméno zařízení a další údaje). Vytváření jednotlivých náhledů probíhá v cyklu.

Mechanismy uvnitř generující metody se pokouší připojit k získané IP adrese a podle dostupnosti generují obsah záznamu. U dostupných RB je zobrazen údaj o počtu připojených uživatelů. Při zobrazení bližších informací záleží opět na stavu zařízení. Zobrazení informací zajišťují třídy `router_info_offline` a `router_info_online`. Posledně jmenovaná generuje rozsáhlejší informace, je však použitelná, jen u dostupných zařízení. Vygenerováno je i zobrazení statické mapy s pozicí RB.

Název zařízení: MikroTik Přístupová IP adresa: 10.10.1.3 Funkce: Přístupový bod Wifi Počet připojených zařízení: 0 Počet registrovaných zařízení: 3 Status: online Bližší informace
Název zařízení: AP1 Přístupová IP adresa: 10.10.1.5 Funkce: Přístupový bod Wifi Status: offline Bližší informace

Obrázek 5.6: Rychlý náhled na registrovaná zařízení.

Aktivace připojení

Připojení se stává pro systém aktivní, ve chvíli kdy mu je přiřazeno jakékoliv registrované rozhraní. U neaktivního připojení je hodnota rozhraní rovna nule.

Aktivovat lze jen neaktivní připojení registrované v systému. K registraci připojení slouží stránka `connect_new`. Registrace je úspěšně provedena jen s dosud neevidovanou MAC adresou. Připojení je po vytvoření vždy neaktivní. Aktivaci připojení je poté možné provést na stránce `connect_activation`.

Při aktivaci je možné zadat statickou adresu (zadáva uživatel) nebo ponechat volbu IP adresy na systému. Systém si uchovává poslední dynamicky přidělenou IP adresu u každého DHCP serveru a přiděluje následující možnou adresu. Může se stát, že jsou všechny IP adresy již vyčerpané, pak klient nemůže být připojen. I statické adresy musí být v rozsahu přidělovaných DHCP adres. Pokud uživatel poprvé zadá IP adresu, která je mimo rozsah, je mu rozsah adres vypsan. Důležité je také nastavit přípojně rozhraní. Vždy jsou nabízena pouze rozhraní stejného typu.

Z neaktivovaných záznamů generuje objekt `activation_connect` tabulku (co záznam, to řádek). Aby bylo možné vždy rozpoznat, který záznam má být aktivován, jsou záznamy tvořeny pomocí samostatného formuláře. Hodnoty v takto odeslaném formuláři jsou rozlišeny na základě MAC adresy poslané přes skryté pole (HTML tag `hidden`). Samotnou aktivaci připojení zajišťuje třída `mac_activate` a metoda `generate`, která zde výjimečně negeneruje HTML obsah. Tato metoda nejdříve zkontroluje, jestli je zařízení, ke kterému má být připojení realizováno dostupné, dále zkontroluje, zda IP adresa patří do rozsahu přidělovaných DHCP adres. Pokud se jedná o bezdrátové rozhraní, je MAC adresa přidána do seznamu povolených připojení (access list) daného rozhraní. Už nezávisle na typu rozhraní je IP adresa svázána s MAC adresou v seznamu vypůjčených adres. Nakonec je v ROS vytvořen záznam typu `simple queue`, který umožňuje omezení rychlosti připojení a zobrazení grafu. Po úspěšně provedené operaci dioda na zařízení krátce zabliká.

68:EB:AE:47:6A:90	Wifi Samsung Monte	ondra.lestinsky@seznam.cz	Statická adresa:	ANO	10.10.21.120	AP1: wlan1	Aktivovat	Smazat
AA:BB:CC:DD:EE:FF	Wifi František Novák	novak@seznam.cz	Statická adresa:	NE		MikroTik: wlan1	Aktivovat	Smazat

Obrázek 5.7: Aktivace připojení.

Deaktivace připojení

Aktivované připojení lze buď deaktivovat, anebo přímo odstranit ze systému. Deaktivace probíhá na stránce `connect_deactivation`. Aktivované záznamy jsou uspořádány do tabulky a seřazeny podle jména RB. Řádky tabulky generuje třída `connect_mac_acces`. V případě, že RB není dostupný, jsou připojení na něj vázaná zobrazena bez možnosti provedené operace. U každého připojení je zobrazen údaj o aktuálním stavu (online nebo offline). Při zavolání metody `mac_deactivate` dojde k vyjmutí záznamu ze seznamu povolených připojení (odebrání přidělené IP adresy, v databázi je přístupové rozhraní nastaveno na hodnotu 0). Je možné tedy deaktivovat jen připojení vázané k dostupným RB.

Editace připojení

K editování informací slouží stránka `connect_edit`. Je možné upravit jméno a email (třída `edit_connection_list`, metoda `edit`). Ve formuláři jsou zobrazeny i údaje o aktuálním stavu připojení. Zobrazeny jsou zobrazeny aktivní i neaktivní připojení.

Pokud je zařízení, které zajišťuje připojení klienta dostupné, může uživatel omezit rychlost připojení (stahování i posílání). Pro tento účel jsou vyhrazeny dva výběrové seznamy obsahující předdefinované hodnoty rychlostí. Po odeslání formuláře je rychlost omezena pomocí `simple queue` (vlastnost `max-limit`) [13]. Omezení rychlosti provede metoda `API_set_simple_queue_limit`. Rychlost odesílání a přijímání dat na sobě nezávisí. V parametrech jsou specifikovány obě hodnoty zvlášť.

Zobrazení všech připojení

Seznam všech připojení je možné zobrazit na stránce `connect_show`. Pro zobrazení je použito uživatelské jméno administrátora, které zajišťuje zobrazení kompletního seznamu všech registrovaných připojení. Jde o téměř stejnou stránku jako pro deaktivaci připojení. U záznamů však nejsou přístupné ovládací tlačítka (Obr. 5.8). U aktivních a zároveň dostupných bezdrátových klientů je zobrazena i úroveň signálu a rychlost linky.

AP1 - 10.10.1.5							online
68:EB:AE:47:6A:90	Samsung Monte	ondra.lestinsky@seznam.cz	wlan1	10.10.21.120	-85dBm@1Mbps		online
C4:46:19:72:41:36	Miroslav Mendroš	mendor@google.com	wlan1	10.10.21.101			offline
MikroTik - 10.10.1.3							online
00:1E:33:4A:70:EC	Ondra LAN	ondra.lestinsky@seznam.cz	ether1	10.10.1.1			online
00:1F:3C:7A:BF:47	Ondra Wifi	ondra.lestinsky@seznam.cz	wlan1	10.10.10.3			offline
Neaktivovaná připojení							offline
AA:BB:CC:DD:EE:FF	František Novák	novak@seznam.cz			DHCP		offline

Obrázek 5.8: Seznam připojení.

Zobrazení statistik

Přístup ke statistikám dovoluje stránka `stat_router`. Stránka při vstupu obsahuje jen výběrový seznam se jmény všech dostupných RB. Po odeslání uživatel vybere typ grafu, na který se chce podívat. Až poté mu je zobrazena čtveřice grafů (denní, týdenní, měsíční, roční). Všechny grafy jsou uloženy v ROS, proto se grafy adresují pomocí IP adresy zařízení a cesty ke grafu. Adresace týdenního grafu, který zobrazuje úroveň vytížení CPU:

`IP_adresa/graph/cpu/weekly.gif`

Adresovat lze i graf, ve kterém uživatel může sledovat vytížení jednotlivých rozhraní. Příklad pro zobrazení roční statistiky rozhraní `ether1`:

`IP_adresa/graphs/iface/ether1/yearly.gif`

Druhá stránka, na které je možné prohlížet grafy je `stat_connect`. Jednotlivé grafy se adresují pomocí jména uvedeného v `simple_queue`. Jednotlivé fronty (`queue`) jsou pojmenovány podle identifikátoru připojení—MAC adresou. Adresace je v tomto případě komplikovanější, při adresaci front nesmí být použit znak „:“, který odděluje hodnoty v MAC adrese. V síťové adresaci je tento znak, vyhrazen pro oddělení čísla portu, proto ROS nahrazuje dvojtečku v názvu fronty znaky „%3A“. V PHP existuje funkce `ereg_replace`, která náhradu bez problémů zvládá. Denní graf pro MAC adresu `AA:BB:CC:DD:EE:FF` je adresován následovně:

`IP_adresa/graphs/queue/AA%3ABB%3ACC%3ADD%3AEE%3AFF/daily.gif`

5.8 Metriky kódu

Počet souborů: 67

Počet tříd: 32

Počet řádků kódu: 12015

Velikost zdrojového textu: 334,3 KB

Kapitola 6

Závěr

V této závěrečné kapitole představím na jednoduché síti základní práci se systémem a hodnotím implementované funkce. Na úplný závěr se podíváme na možná rozšíření, která by do systému mohla být v budoucnu přidána.

6.1 Demonstrace navrženého systému

Pro testování systému jsem měl k dispozici dvě zařízení Mikrotik (RouterBoard 411AH, ROS verze 4.17). Tento typ má jediné ethernetové rozhraní a jeden mini-pci slot, který byl osazen bezdrátovým adaptérem.

Pro ověření funkčnosti systému jsem vytvořil jednoduchou síť, ve které jsou ethernetová rozhraní připojena do centrálního přepínače (switch), ke kterému je připojen i počítač s informačním systémem. Pro plnou funkčnost systému (zobrazení map) je ke switchy připojena i brána zajišťující přístup do internetu. Všechna rozhraní u aktivních prvků sítě (RB, server s IS, brána) mají nastavenou statickou IP adresu. Informační systém se tak může připojit ke všem zařízením. Celkově je popisovaná síť rozdělena na tři podsítě (páteří a pro dvě u bezdrátových rozhraní) a internet. Pomocí směrovacího protokolu RIP je zajištěna dostupnost všech podsítí. Bezdrátová rozhraní jsou zabezpečena klíčem WPA PSK.

Při uvedené konfiguraci sítě se do IS může nejdříve přihlásit hlavní administrátor jenom ze stanice, na které je IS nasazen. Po jeho přihlášení do systému má možnost vytvořit další uživatelské účty a nastavit k nim patřičná oprávnění. Možností jak tato oprávnění nastavit je celá řada a záleží jen na hlavním správci, co uživatelům umožní. Nejčastěji uživatel bude oprávněn ke správě několika zařízení a k nim realizovaným připojením včetně tvorby nových připojení. Administrátor může například vytvořit i uživatele, který má na starosti pouze připojování a odpojování klientů u jediného vybraného zařízení, stejně tak je možné, nechat uživateli téměř shodná oprávnění jako má hlavní administrátor.

V popisované topologii je možné do systému zaregistrovat oba připojené RB. Při jejich registraci přihlášený uživatel zvolí jméno zařízení, popis jednotlivých rozhraní a umístění zařízení v mapě. Od tohoto okamžiku je dostupná konfigurace jednotlivých rozhraní. V případě, že se změní IP adresa (IP adresu může změnit administrátor pomocí konfiguračních nástrojů) již registrovaného zařízení, nepůjde se znovu k zařízení připojit a zařízení bude muset být odebráno ze systému, který možnost změny IP adresy rozhraní nemá dovolenou.

Všechna bezdrátová rozhraní uvedená v popisovaném schématu je nutné aktivovat, nastavit SSID, zapnout MAC autentizaci a zvolit vysílací kanál. Aby bylo možné připojovat uživatele, je potřeba zapnout DHCP server. U DHCP serveru je nastavena přidělovaná

výchozí brána (adresa rozhraní RB), DNS server a přidělovaný rozsah IP adres.

Od této chvíle je systém připraven pro správu klientů. Klientova MAC adresa spolu s jeho jménem je nejdříve zaevidovaná v systému a až poté je aktivovaná u zvoleného RB. Po jeho aktivaci klient získává přístup do částí, které mu jsou dostupné (systém toto neřeší). Uživatelé systému se nyní mohou do systému přihlásit z jakékoliv stanice připojené v síti, avšak pokud je přihlášený uživatel připojený pomocí bezdrátového rozhraní, musí si dát pozor, aby si v systému nedeaktivoval své připojení a nezablokoval si tím přístup do sítě. Pokud klient potřebuje změnu přístupového bodu, je nutné klientovo připojení nejprve deaktivovat a poté znovu aktivovat u nového rozhraní. V případě, že by některý z připojených klientů nadměrně zatěžoval síť a omezoval tím ostatní klienty, je možné omezit rychlost jeho připojení.

Při výpadku části sítě si uživatel může v systému zobrazit mapu všech zařízení a pomocí ní zjistit, která zařízení jsou nedostupná a reagovat na tuto situaci. V případě, že je systém bez přístupu k internetu, může uživatel zkontrolovat dostupnost pomocí rychlého náhledu na všechna registrovaná zařízení. Pokud je výpadek způsoben poškozením RB, je možné ho nahradit stejným typem, na který je přeneseno nastavení stávajícího zařízení.

6.2 Zhodnocení systému

Cílem této práce bylo vytvořit webový informační systém, který centrálně spravuje a monitoruje zařízení se systémem RouterOS. Na základě tohoto požadavku byl navrhnut systém, ve kterém je nutné zařízení nejprve zaregistrovat a až poté je umožněna jeho správa. U registrovaných zařízení je možné zobrazit jejich aktuální stav a základní informace o zařízení.

Systém bezproblémově zvládá správu bezdrátových klientů, u kterých dokáže omezit jejich přístup a zobrazit informace o kvalitě signálu, dále umožňuje evidenci klientů, včetně uchování jejich IP adres, které sám umí přidělit. Navíc lze zobrazit graf s klientským datovým tokem, na jehož základě je možné omezit rychlost připojení jednotlivých klientů.

Požadavek na správu administrátorských rolí v jednotlivých částech sítě je vyřešen individuálním povolením přístupu uživatele k zařízení. Uživatelský účet je rozšířen o seznam povolených funkcí v systému. Přístup do systému je chráněn pomocí uživatelského jména a hesla.

Systém dokáže zobrazit všechny dostupné grafy, které ROS nabízí. Spolu s grafy systém dokáže zobrazit i mapy, používá k tomu aplikaci Google Map. V mapě lze zobrazit jak polohu jednotlivých zařízení, tak i kompletní mapu sítě včetně aktuálního vizuálně rozlišeného stavu dostupných zařízení.

6.3 Vylepšení do budoucna

Zápis konfigurace do ROS je v některých chvílích zdlouhavý, proto by do budoucna nebylo špatné více zoptimalizovat jednotlivé konfigurační příkazy a dobu čekání tak zkrátit. Konfiguraci ROS by také bylo možné v periodických cyklech ukládat do databáze, aby nemohlo dojít k nekonzistenci dat. U uživatelů lze specifikovat oprávnění jen na úrovni blokování celého menu, přístupová oprávnění by mohla být omezena přímo na jednotlivé služby (nastavení DHCP serveru, povolení statické adresy u klienta apod.).

Spojení systému s mapovými podklady v sobě nese obrovský potenciál. Mapa může zobrazit na základě směrovacích protokolů propojení uzlů sítě. Úplně od věci by nemuselo

být zobrazení přibližné polohy připojeného klienta na základě síly jeho signálu. Systém může zobrazovat i rozšířené grafy, například propojením s aplikací **Cake**.

Literatura

- [1] *RouterBOARD* [online]. 2011 [cit. 2011-05-4]. Dostupné na:
<http://routerboard.com/?group_id=11>.
- [2] ACHOUR, M. et al. *PHP manual* [online]. 2011 [cit. 2011-05-4]. Dostupné na:
<<http://code.google.com/intl/cs/apis/maps/>>.
- [3] BURGET, R. a BURGETOVÁ, I. *Kaskádové styly (CSS)* [online]. 2011, Aktualizováno 11.2.2010 [cit. 2011-05-4]. Dostupné na:
<<https://www.fit.vutbr.cz/study/courses/ITW/private/prednasky/p02/index.html>>.
- [4] BURGET, R. a BURGETOVÁ, I. *Značkovací jazyky* [online]. 2011, Aktualizováno 11.2.2010 [cit. 2011-05-4]. Dostupné na:
<<https://www.fit.vutbr.cz/study/courses/ITW/private/prednasky/p02/index.html>>.
- [5] GOOGLE CODE. *Google Maps API Family* [online]. 2011 [cit. 2011-05-4]. Dostupné na: <<http://code.google.com/intl/cs/apis/maps/>>.
- [6] GOOGLE CODE. *Google Maps JavaScript API V3* [online]. 2011 [cit. 2011-05-4]. Dostupné na:
<<http://code.google.com/intl/cs/apis/maps/documentation/javascript/>>.
- [7] GOOGLE CODE. *Static Maps API V2 Developer Guide* [online]. 2011 [cit. 2011-05-4]. Dostupné na: <<http://code.google.com/intl/cs/apis/maps/>>.
- [8] MIKROTIK. *Routers and Wireless* [online]. 2011, Last-Modified: Wed, 04 May 2011 [cit. 2011-05-4]. Dostupné na: <<http://www.mikrotik.com>>.
- [9] MIKROTIK DOCUMENTATION. *Graphing* [online]. 2011, revision: February 6 2008 [cit. 2011-05-4]. Dostupné na:
<<http://www.mikrotik.com/testdocs/ros/3.0/aaa/graphing.php>>.
- [10] MIKROTIK WIKI. *API PHP class* [online]. 2011, modified on 23 March 2011 [cit. 2011-05-4]. Dostupné na: <http://wiki.mikrotik.com/wiki/API_PHP_class>.
- [11] MIKROTIK WIKI. *Manual:API* [online]. 2011, modified on 13 April 2011 [cit. 2011-05-4]. Dostupné na: <<http://wiki.mikrotik.com/wiki/Manual:API>>.
- [12] MIKROTIK WIKI. *Manual:License levels* [online]. 2011, modified on 13 April 2011 [cit. 2011-05-4]. Dostupné na:
<http://wiki.mikrotik.com/wiki/Manual:License_levels>.
- [13] MIKROTIK WIKI. *Manual:Queue* [online]. 2011, modified on 14 April 2010 [cit. 2011-05-4]. Dostupné na: <<http://wiki.mikrotik.com/wiki/Manual:Queue>>.

- [14] MySQL. *MySQL Editions* [online]. 2010 [cit. 2011-05-4]. Dostupné na: [<http://www.mysql.com/products/>](http://www.mysql.com/products/).
- [15] SCHLOSSNAGLE, G. *Pokročilé programování v PHP5*. Přel. R. Řežábek. 1. vyd. Koželužská 7, Brno: Zoner Press, 2004. 624 s. ISBN 80-86815-14-5.
- [16] W3SCHOOLS. *The XMLHttpRequest Object* [online]. 2011 [cit. 2011-05-4]. Dostupné na: [<http://www.w3schools.com/XML/xml_http.asp>](http://www.w3schools.com/XML/xml_http.asp).
- [17] ŠMÍD, V. *Pojem informačního systému* [online]. 2011, změněno Čt 9. leden 2003 [cit. 2011-05-4]. Dostupné na: [<http://www.fi.muni.cz/~smid/mis-infsys.htm>](http://www.fi.muni.cz/~smid/mis-infsys.htm).

Dodatek A

Obsah CD

Na přiloženém CD jsou umístěny zdrojové texty a manuál.